High-Speed Network Communication on the Cray XT

Kitrick Sheets
n8851@cray.com

CRAY
THE SUPERCOMPUTER COMPANY

Agenda

- XT communication overview
- Portals overview
- Interconnect specifics
- Interesting communication patterns
- Wrap-up

CRAY
THE SUPERCOMPUTER COMPANY

XT Communication Stack

Additional APIs

MPI | shmem | GA

Portals API

DVS | pkAPI | TCP/IP

Portals NAL

Lustre

BEER | SSIP

Seastar NIC

...to XT mesh

User

Kernel

Hardware/Firmware

CRAY
THE SUPERCOMPUTER COMPANY
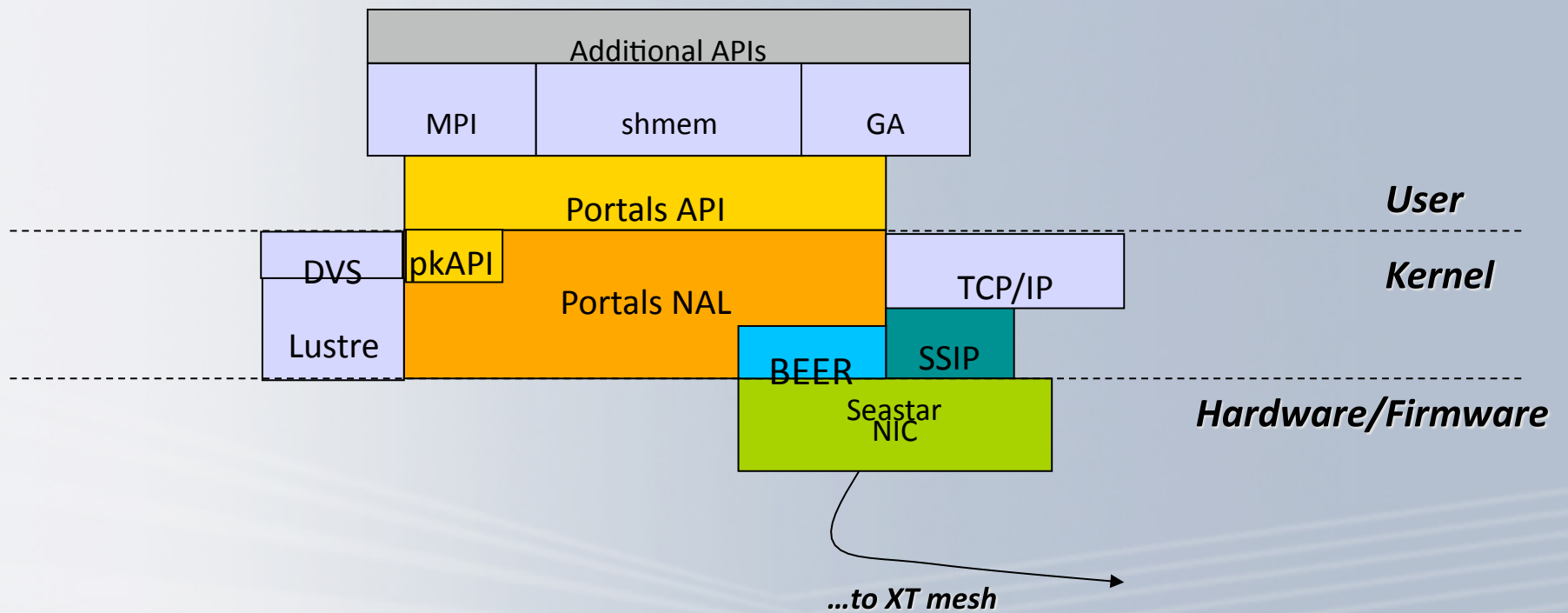
Portals

- Network communication API developed by Sandia
  - Data movement is between 'portal's into user address spaces
  - Supports zero-copy, OS bypass, and Application bypass
  - Supports one-sided communication (Get and Put)

- Primary method of communication on XT interconect
  - Cray extensions for:
    - CLE optimization
    - Interrupt management improvement
    - Multi-core nodes
    - Atomic operations (GetPut, GetAdd, CGetPut)
    - Network resiliency (Basic End-to-End Reliability)

- Portals Assumptions
  - Network is reliable and deterministic
  - Guaranteed in-order transmission of messages between NID/PID pairs
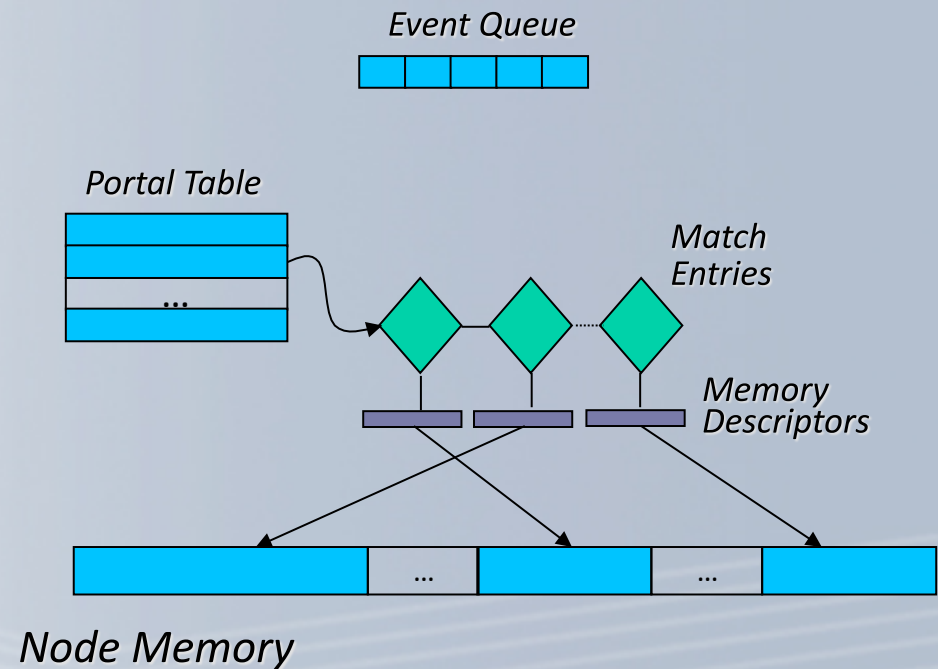
CRAY
THE SUPERCOMPUTER COMPANY

Portals Structure

- Portal Table
  - Anchor table for registered portals users

- Event Queue
  - Completion/notification

- Match Entries
  - Filters for in-bound RDMA operations

- Memory Descriptors
  - Describe memory segments attached to match entries

Event Queue

Portal Table

Match Entries

Memory Descriptors

Node Memory
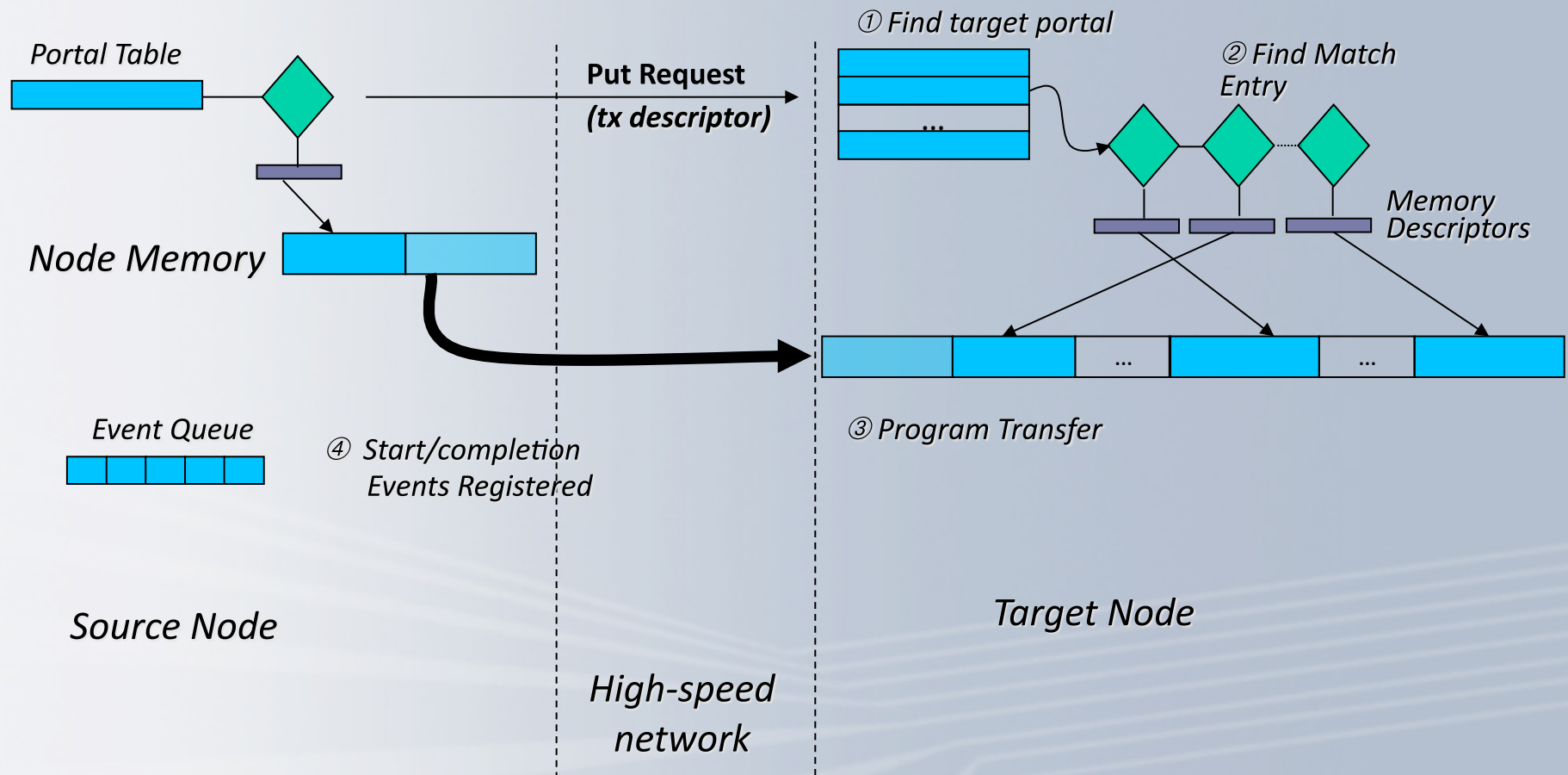
CRAY
THE SUPERCOMPUTER COMPANY
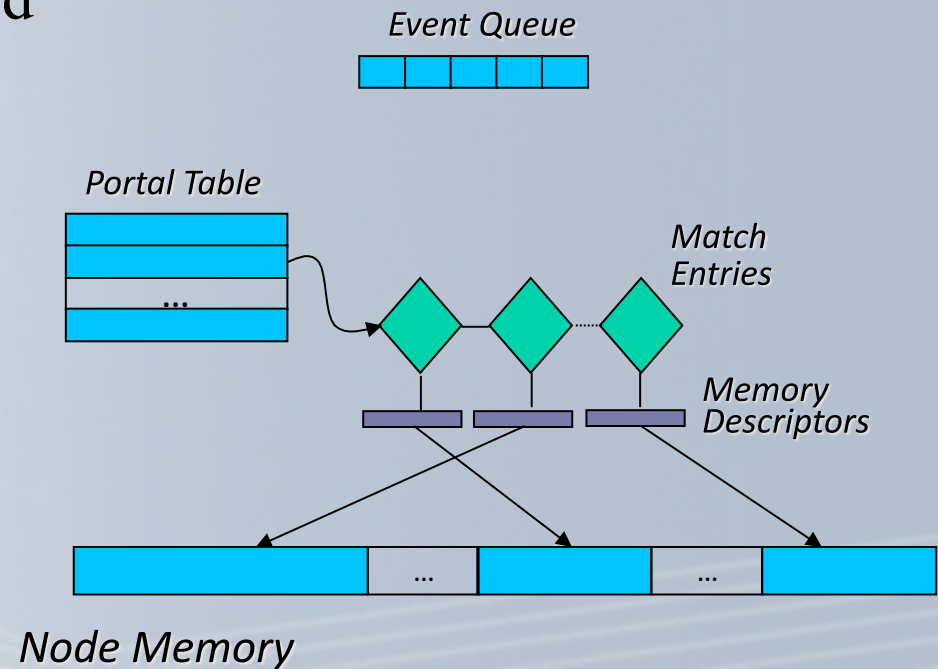
Portals API

- Portal Table
  - PtlNIInit(), PtlFini()

- Event Queue
  - PtlEQAlloc()

- Match Entries
  - PtlMEAttach()

- Memory Descriptors
  - PtlMDAttach(), PtlMDBind()
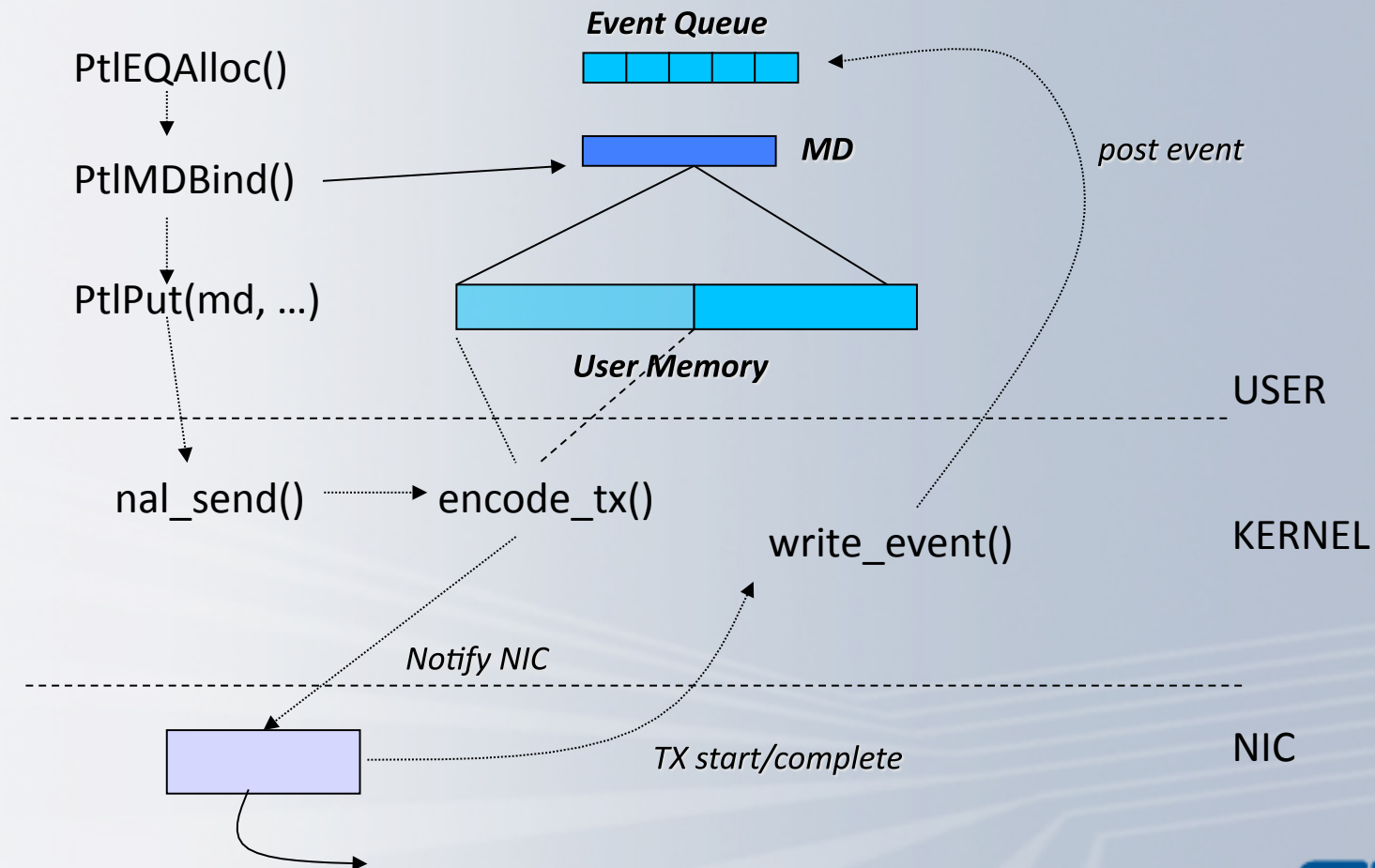
- Data Movement
  - PtlGet(), PtlPut()

Event Queue

Portal Table

Match Entries

...

Memory Descriptors

...   ...

Node Memory

CRAY
THE SUPERCOMPUTER COMPANY

Portals Data Transfer

Portal Table

*Put Request
(tx descriptor)*

① *Find target portal*

② *Find Match
Entry*

...

*Memory
Descriptors*

Node Memory

Event Queue

④ *Start/completion
Events Registered*

...   ...

③ *Program Transfer*

Source Node

Target Node

*High-speed
network*

CRAY
THE SUPERCOMPUTER COMPANY

Seastar Network Access Layer

- Provides bridge between portals API and NIC
- Manages transmit and receive state for in-flight portals requests
- Works in conjunction with portals library for message validation and matching
- Translates logical requests into DMA programs to be consumed by the NIC
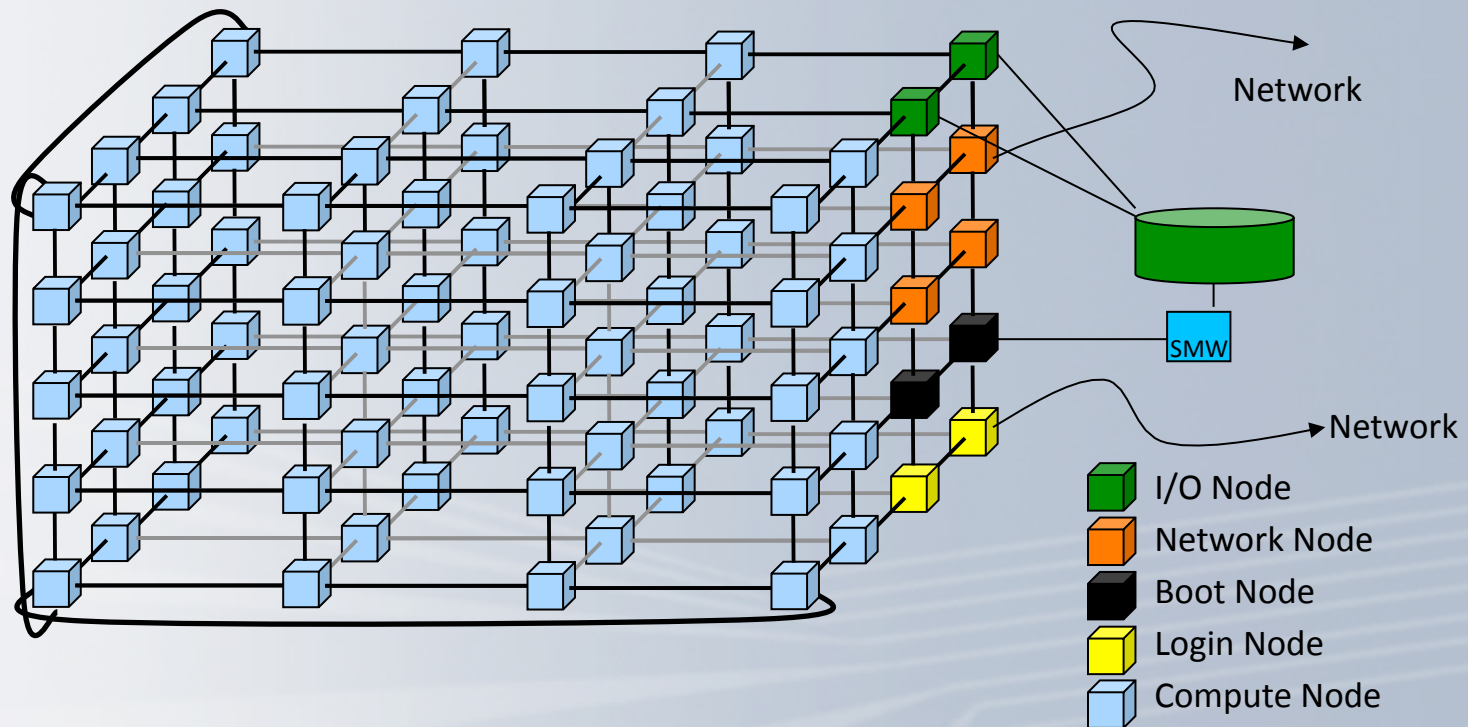
CRAY
THE SUPERCOMPUTER COMPANY

User Transmit Request Flow

Event Queue

PtlEQAlloc()

PtlMDBind() → **MD**

post event

PtlPut(md, …)

User Memory

USER

nal_send() ⋯→ encode_tx()

write_event()

KERNEL

Notify NIC

TX start/complete

NIC

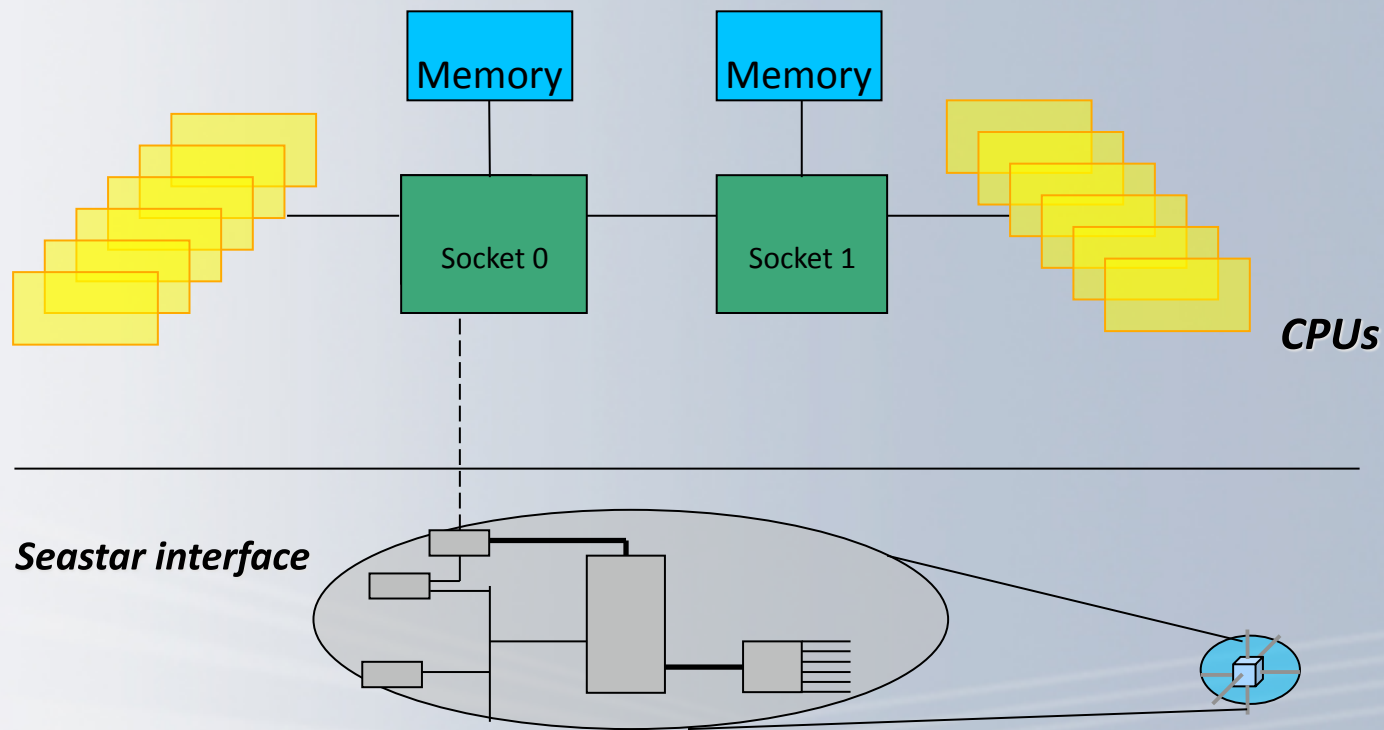Page 9

CRAY
THE SUPERCOMPUTER COMPANY

Providing reliable communication

- Basic End-to-End Reliability (BEER) protocol
- Host-resident software layer between NIC and portals
- Provides recovery for various message transmission issues
  - CAM overflow
  - OS resource exhaustion
  - Flow control for tx/rx imbalance and forward progress
  - Cleanup for orphaned transmits (remote node failures, etc.)
  - Warm reboot of nodes
- Message sequence numbers used to manage flow between NID/PID pairs

CRAY
THE SUPERCOMPUTER COMPANY

XT System Architecture

Network

SMW

Network

- ■ I/O Node
- ■ Network Node
- ■ Boot Node
- ■ Login Node
- ■ Compute Node

CRAY
THE SUPERCOMPUTER COMPANY

Seastar connection to CPUs

Memory

Memory

Socket 0

Socket 1

CPUs

Seastar interface

Network Interface - Seastar

PPC

Packet DMA Engine

Router

HT2LB

HT

Compute complex
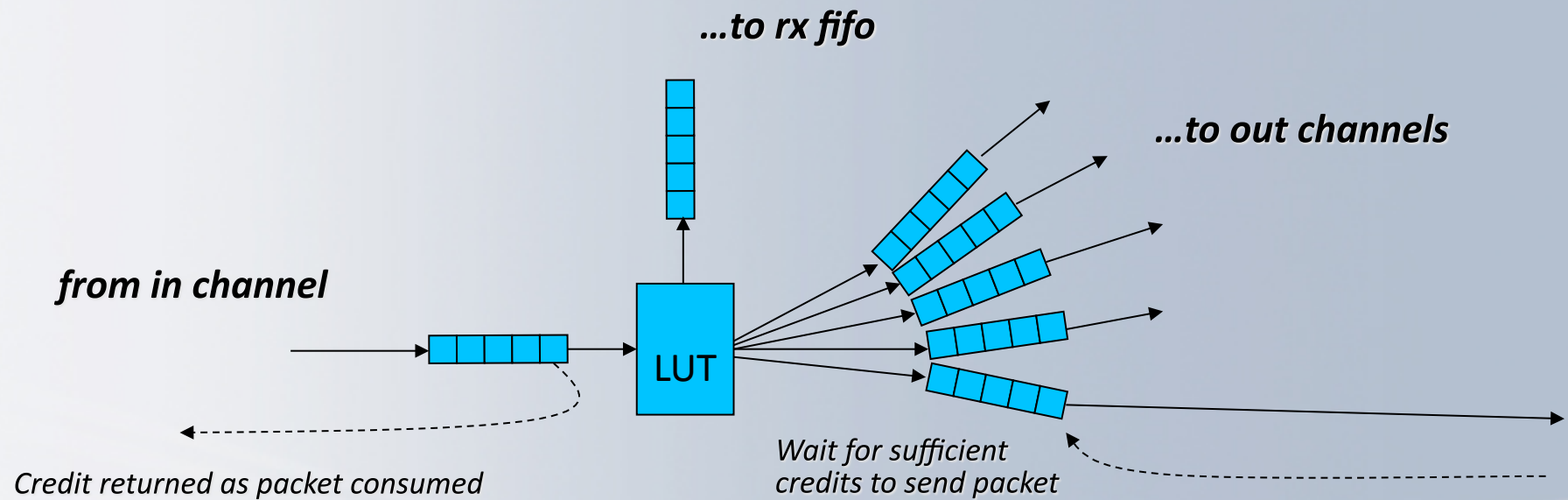
HT

CRAY
THE SUPERCOMPUTER COMPANY
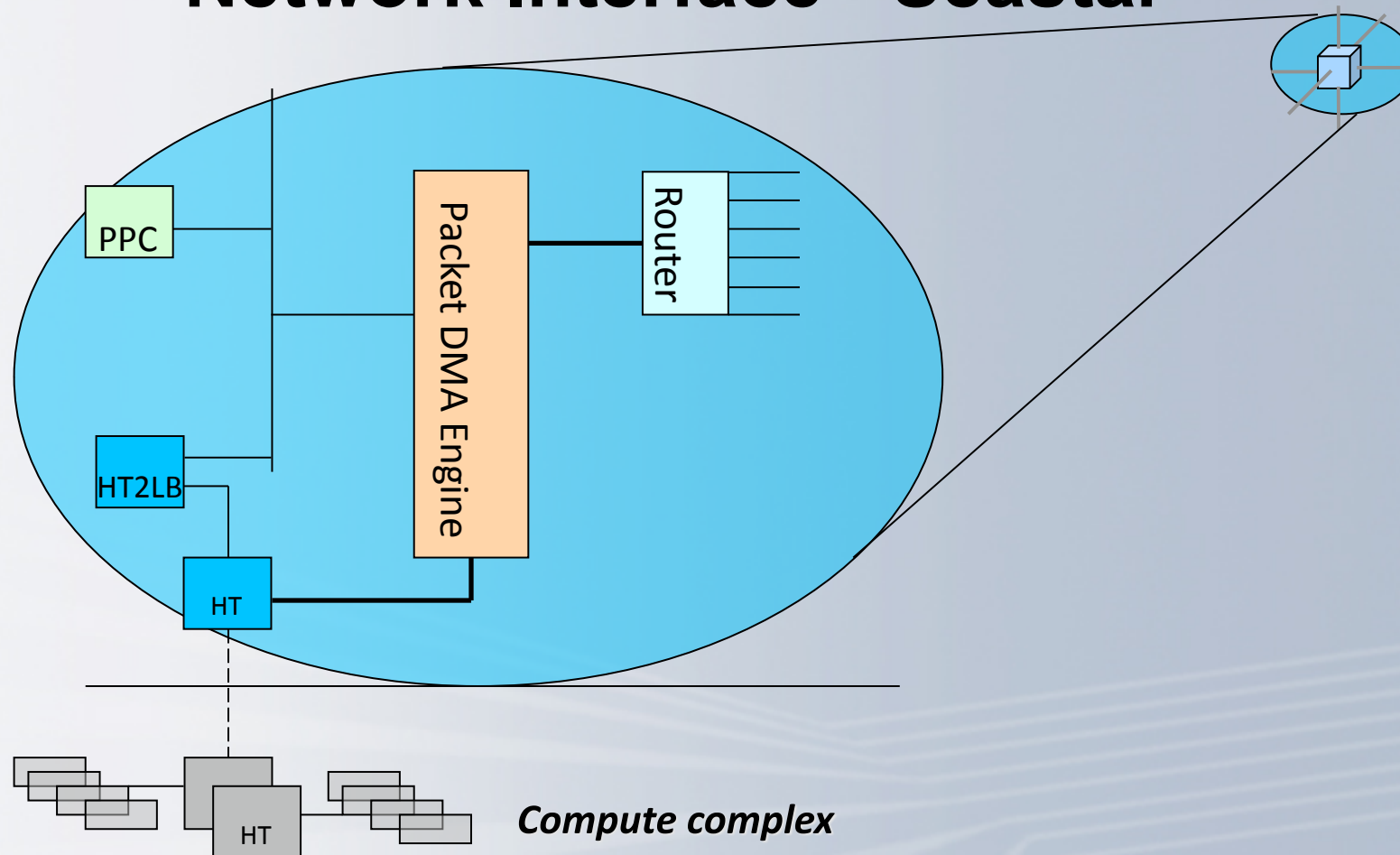
Embedded Processor Complex

- PPC 440 Processor

- 384KB on-board RAM

- Manages DMA engine
  - Interrogates incoming requests and coordinates with host for data placement
  - Keeps tx engine fed

- Provides RAS functionality
  - Warm reboot
  - Host failure recovery
  - Maintenance interface communication
  - Seastar fault detection and recovery

CRAY
THE SUPERCOMPUTER COMPANY

Seastar Router

- Connects the seastar to the 3D torus network
  - 6 network ports
  - 1 host port
- Supports up to 32K nodes
- Each port supports 3.25GB/s bandwidth
- Supports cut-through routing
- Basic flow control unit is a flit
  - Flit contains 64 data bits + control bits
- Flits are combined into network packets
  - Packets consist of a header flit and up to 8 data flits
- Hardware flow control is credit based
  - Credits are replenished when receiving router accepts flit

CRAY
THE SUPERCOMPUTER COMPANY

Seastar Router

...to rx fifo

...to out channels

from in channel

LUT

Credit returned as packet consumed

Wait for sufficient credits to send packet

LUT = Lookup Table: routes inbound request through switch to destination channel
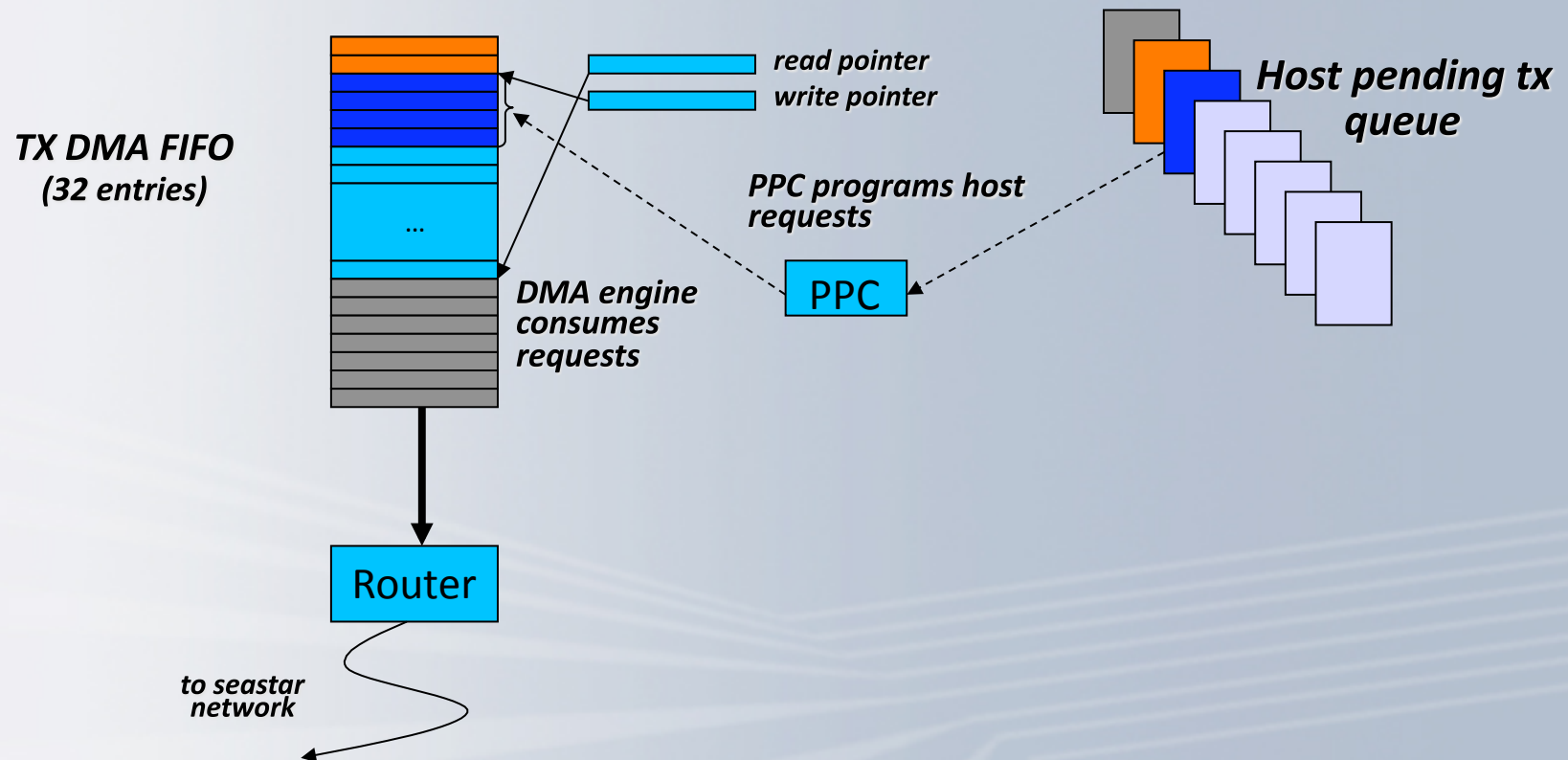
Seastar DMA Engine

- **Transmit Engine**
  - Single transmit DMA engine
  - Responsible for translation of outbound block transfers into network flits
  - PPC manages 32 entry TX DMA queue
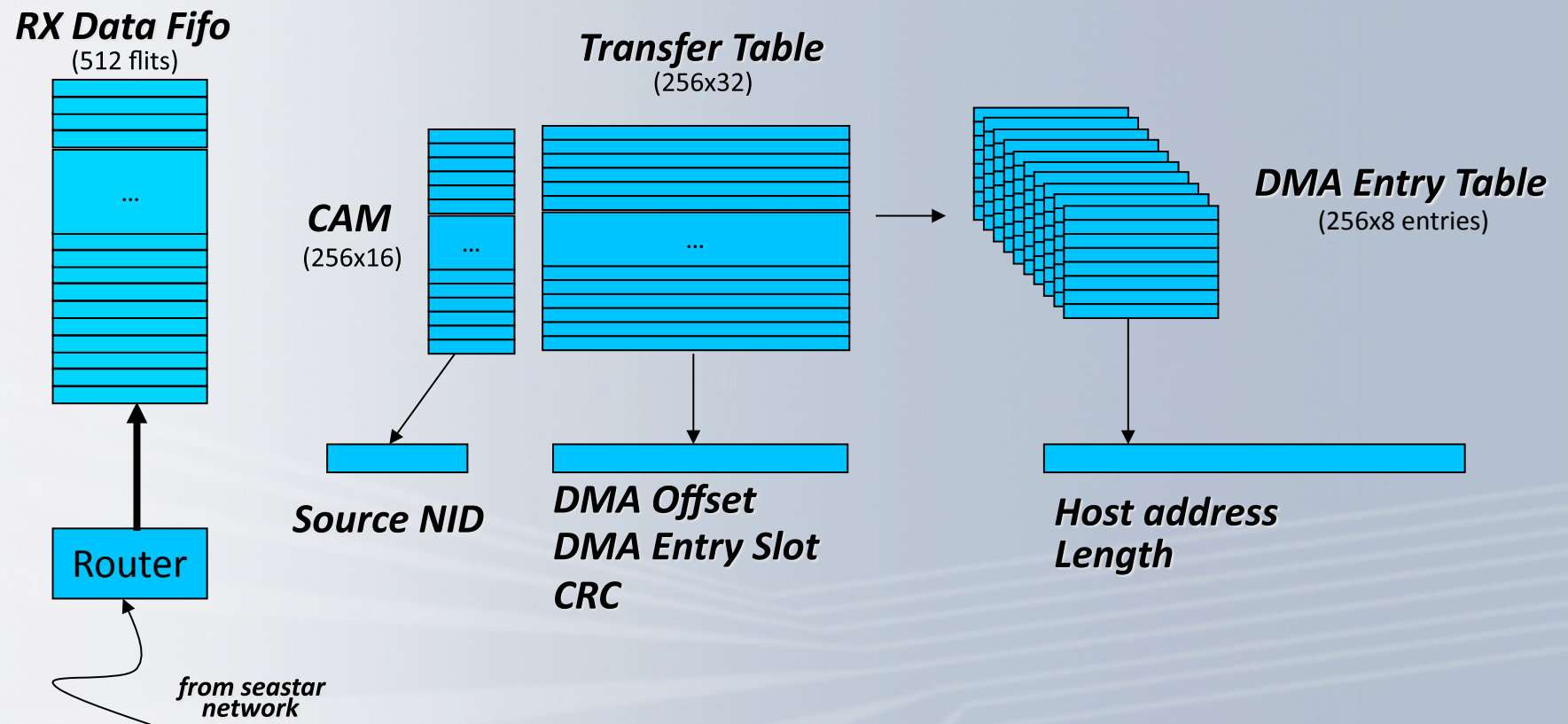 - Price of tx = 1 credit / flit

- **Receive Engine**
  - Fed by 512 flit rx data fifo
  - Content Addressable Memory (CAM) maps remote source to each of the 256 rx queues
  - On CAM match, packet is sent to matching rx DMA stream
  - Router credited when packet is moved from rx data fifo

CRAY
THE SUPERCOMPUTER COMPANY

Seastar Transmit Machinery

TX DMA FIFO
(32 entries)

…

read pointer
write pointer

DMA engine consumes requests

PPC programs host requests

PPC

Host pending tx queue

Router

to seastar network

CRAY
THE SUPERCOMPUTER COMPANY

Seastar Receive Machinery

RX Data Fifo
(512 flits)

CAM
(256x16)

Transfer Table
(256x32)

DMA Entry Table
(256x8 entries)

Source NID

DMA Offset
DMA Entry Slot
CRC

Host address
Length

Router

from seastar network

CRAY
THE SUPERCOMPUTER COMPANY
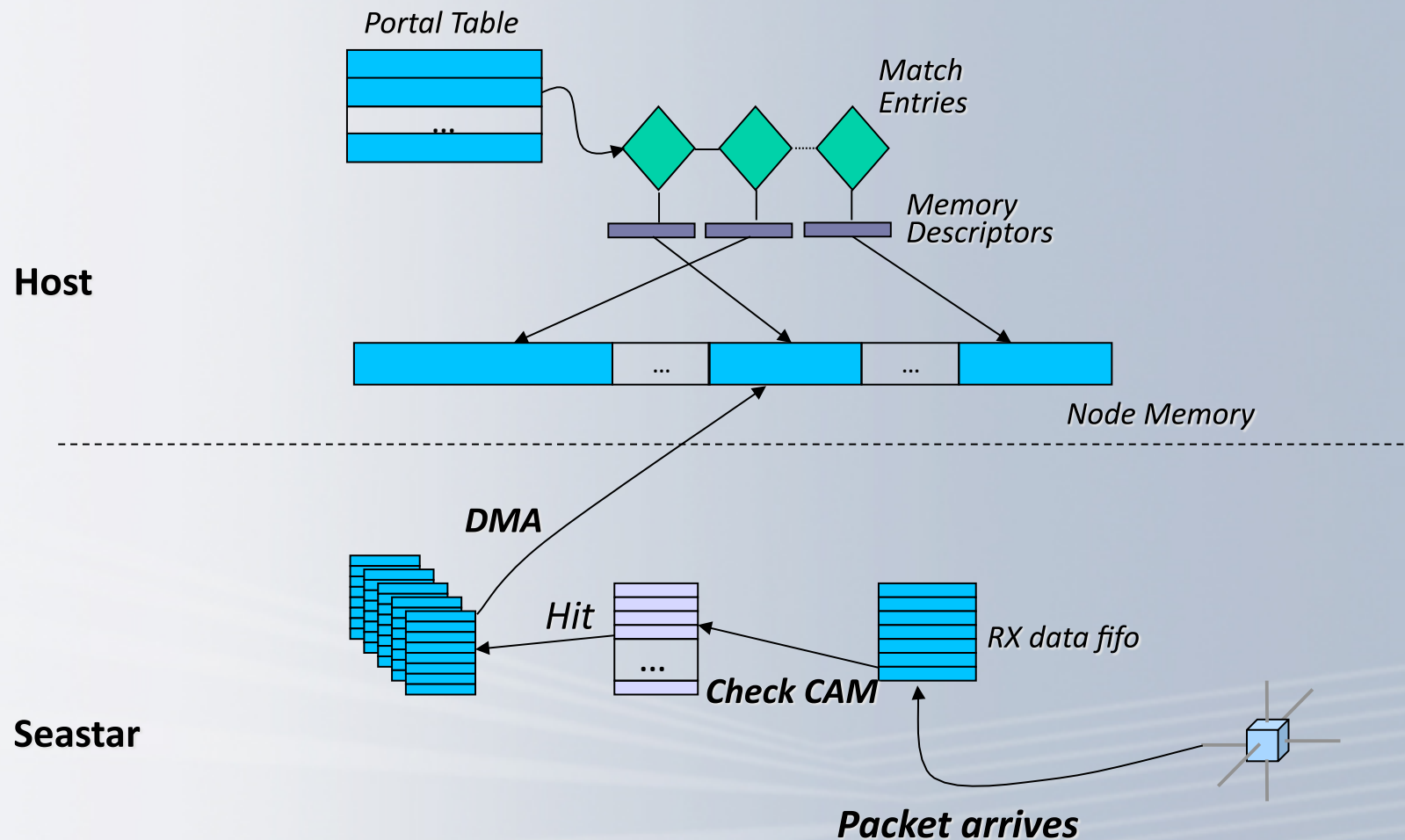
Key DMA Parameters

- **TX DMA**
  - Single transmit FIFO
  - Requests are consumed in the order presented
  - PPC responsible for feeding host requests into engine
  - Transmit possible only when credits available at target router
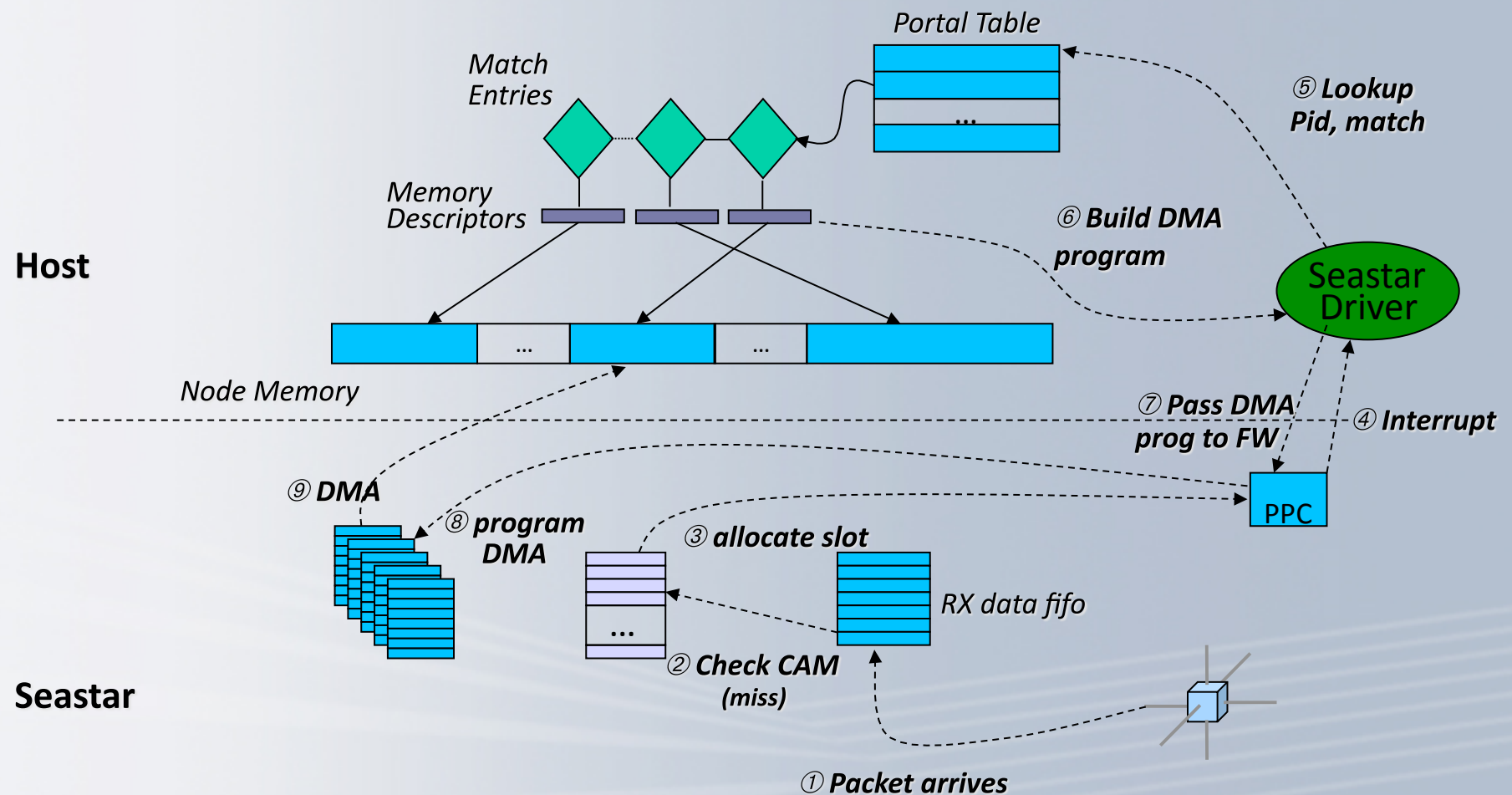
- **RX DMA**
  - Single receive FIFO
  - Small messages (<= 16 bytes) handled directly from FIFO
  - Large messages require CAM for DMA
  - PPC/host intervene on CAM miss
  - Router credits returned when packet moves from FIFO

Normal message reception

Portal Table

Match Entries

Memory Descriptors

Host

Node Memory

DMA

Hit

Check CAM

RX data fifo

Seastar

Packet arrives

CRAY
THE SUPERCOMPUTER COMPANY

CAM Miss

Portal Table

Match
Entries

⑤ *Lookup*
Pid, match

Memory
Descriptors

⑥ *Build DMA*
program

Host

Seastar
Driver

Node Memory

⑦ *Pass DMA*
prog to FW

④ *Interrupt*

PPC

⑨ *DMA*

⑧ *program*
DMA

③ *allocate slot*

RX data fifo

Seastar

② *Check CAM*
(miss)

① *Packet arrives*

CRAY
THE SUPERCOMPUTER COMPANY

CAM Overflow

Portal Table

Match Entries

Memory Descriptors

Host

⑤ **Note source Node (BEER)**

Seastar Driver

Node Memory

⑥ **Issue NAK (BEER)**

④ **Interrupt**

⑨ **DMA**

③ **CAM full**

⑧ **Drain msg**

PPC

RX data fifo

⑦ **Queue NAK for TX**

② **Check CAM (miss)**

TX fifo

Seastar

① **Packet arrives**

CRAY
THE SUPERCOMPUTER COMPANY

Problem Scenario 1 – CAM thrash

- All-to-one
  - Actually, > 256 to one

- Effective throughput of the seastar decreases in direct proportion to the sender/receiver ratio
  - CAM overflows induce additional chatter due to:
    - Interrupt/nak sequence
    - Message retransmission
    - Unused data flowing through rx data fifo

- No fairness in CAM allocation scheme
  - CAM slots are 'sticky', but can be resourced at the end of a message.

CRAY
THE SUPERCOMPUTER COMPANY

Scenario 2 – tx/rx imbalance

- All-from-one (e.g., GET storm)
  - Unrelated to CAM issues

- GET requests arrive as single-packet requests, which do not require the CAM
  - Processed directly from rx fifo
  - Not throttled by CAM limit of 256 concurrent sources

- REPLY DMAs fill the tx fifo and requests begin to backup

- BEER initiates flow control when transmit resources become scarce

CRAY
THE SUPERCOMPUTER COMPANY

Scenario 3 – tx interference

- Transmit requests serialized through Seastar
- Large tx stream from one PE can monopolize resources
- Transmit queue is not prioritized
  - Don't initiate many large low-priority (async) requests followed by a small, high priority request (e.g., barrier)
- Must consider asynchronous tx activity
  - Lustre buffer flushes
  - Out-of-band requests (RDMA pulls)

CRAY
THE SUPERCOMPUTER COMPANY

Enhancements for hex-core and beyond

- **CAM Swapping**
  - Firmware enhancements to manage over commitment of CAM
  - Similar in concept to TLB management
  - Eliminates many fault-related retransmissions

- **Efficiency improvements to on-node transfers**
  - Less contention for local resources
  - Reduced back-pressure on network

- **Core specialization**
  - Isolates jitter-related processing (e.g., system services, daemons, etc.)

CRAY
THE SUPERCOMPUTER COMPANY

Summary

- Forward progress on the network is key
  - Stalls/delays at one node can ripple through the network.

- All-to-one communication comes at a price
  - CAM pressure can reduce effective throughput

- Tx and rx communication paths have unique characteristics

- Out-of-band communication can impact application

- Be mindful of communication resource exhaustion costs as application scales

CRAY
THE SUPERCOMPUTER COMPANY